PRACTICAL 3

Topic: Wolf goat cabbage and farmer puzzle

Introduction: I have created a simulation of the wolf cabbage goat farmer game and solved it using rudimentary ai algos

Code:

class RiverCrossingPuzzle:

    def \_\_init\_\_(self):

        # Define the initial state: all items (Farmer, Wolf, Goat, Cabbage) are on the left

        self.left\_side = ["Farmer", "Wolf", "Goat", "Cabbage"]

        self.right\_side = []

        self.boat = []

    def display\_state(self):

        """Display the current state of the river."""

        print(f"Left Side: {self.left\_side}")

        print(f"Right Side: {self.right\_side}")

        print(f"Boat: {self.boat}")

        print("-" \* 40)

    def move(self, item=None):

        """Move an item across the river."""

        # Determine the current location of the boat

        if "Farmer" in self.left\_side:

            source = self.left\_side

            destination = self.right\_side

        else:

            source = self.right\_side

            destination = self.left\_side

        # Move the Farmer and optionally one item

        if "Farmer" in source:

            self.boat.append("Farmer")

            source.remove("Farmer")

        if item and item in source:

            self.boat.append(item)

            source.remove(item)

        # Move the boat to the other side

        destination.extend(self.boat)

        self.boat.clear()

    def is\_safe(self, side):

        """Check if the given side is in a safe state."""

        if "Goat" in side:

            if "Wolf" in side and "Farmer" not in side:

                return False  # Wolf eats Goat

            if "Cabbage" in side and "Farmer" not in side:

                return False  # Goat eats Cabbage

        return True

    def solve(self):

        """Solve the puzzle step by step."""

        # Steps to solve the puzzle

        steps = [

            ("Goat", "Take Goat across"),

            (None, "Return alone"),

            ("Wolf", "Take Wolf across"),

            ("Goat", "Return with Goat"),

            ("Cabbage", "Take Cabbage across"),

            (None, "Return alone"),

            ("Goat", "Take Goat across"),

        ]

        for item, action in steps:

            print(action)

            self.move(item)

            self.display\_state()

            # Check if the current state is valid

            if not self.is\_safe(self.left\_side) or not self.is\_safe(self.right\_side):

                print("Unsafe state detected! Puzzle failed.")

                return False

        print("Puzzle Solved!")

        return True

# Run the puzzle

puzzle = RiverCrossingPuzzle()

puzzle.solve()

Output:

